

2

AN OBJECT-ORIENTED DATA AUTOMATION SYSTEM
FOR
MICROSOFT EXCEL FILES

NG LEE YEAN

DISSERTATION SUBMITTED IN FULFILMENT
OF THE REQUIREMENTS
FOR THE DEGREE OF MASTER OF COMPUTER SCIENCE

FACULTY OF COMPUTER SCIENCE
AND INFORMATION TECHNOLOGY
UNIVERSITY OF MALAYA

KUALA LUMPUR

DECEMBER 2003

Perpustakaan Universiti Malaya



A511203977

ABSTRACT

From our daily business transaction to annual financial reports, Microsoft Excel has been playing a very important role in preparing these reports. Many of the current business activities are documented for future references. Nowadays, Microsoft Excel is widely used for preparing accounts spreadsheets, and recording data in tabular format. The Object-Oriented Data Automation (OODA) System for Microsoft Excel Files is a system that helps Microsoft Excel users to manipulate all the data contained in Excel documents. It is developed using an Object-Oriented programming language, which is Java and running under a Java Servlet Container, Apache Tomcat web server. There are two main features provided in this system. Firstly, the OODA System for Microsoft Excel Files provides an interface for users to segregate an Excel document into several portions according to each pair of starting point and ending point provided. Users could specify words or the Excel cell name as the starting and ending of data segregation. Secondly, this system also provides system automation for combining several Excel documents without starting Microsoft Excel application in order to open each file. Users could browse any Excel documents and specify the cells which they hope to combine with. Furthermore, users could choose to combine the Excel documents horizontally or vertically. An OODA System for Microsoft Excel files is developed for the main purpose of reducing the time taken in manually cut and paste portions of an Excel document. Automation is a critical key in resolving the issue of manipulating large volume Excel documents.

ACKNOWLEDGEMENT

Highest appreciation dedicated to Mr. Chiew Thiam Kian, lecturer from the Faculty of Computer Science and Information Technology, University of Malaya. Mr. Chiew has been giving a lot of guidelines and advices in the whole process of the development of this project. Furthermore, opinions given by Mr. Chiew have improved and corrected many systems bugs and errors.

Besides that, appreciation must also be dedicated to all the system testers. They have contributed their precious time in using this system to segregate or combine their own Microsoft Excel files and filling up the User Acceptance Test forms.

Any party which has directly or indirectly given their support for the completion of this project is highly appreciated as well.

DISCLAIMER

The ownership of this report is reserved by University of Malaya and no part of this dissertation should be reproduced, stored in a retrieval system, or transmitted by any means, electronic, mechanical, photocopying, recording or otherwise without prior written consent from University of Malaya.

TABLE OF CONTENTS

ABSTRACT	II
ACKNOWLEDGEMENT	III
DISCLAIMER	IV
TABLE OF CONTENTS	V
LIST OF FIGURES	IX
LIST OF TABLES	XII
LIST OF ABBREVIATIONS	XIII
LIST OF APPENDIXES	XIV
CHAPTER 1: INTRODUCTION	1
1.1 PREFACE	1
1.2 INTRODUCTION TO THE OBJECT-ORIENTED DATA AUTOMATION SYSTEM FOR MICROSOFT EXCEL FILES	2
1.3 PROBLEM STATEMENT	3
1.4 PROJECT MISSION	5
1.5 OBJECTIVES	5
1.6 PROJECT SCOPES	6
1.7 PROJECT CONTRIBUTIONS	7
1.8 PROJECT ASSUMPTIONS	8
1.9 REPORT ORGANIZATION	8
CHAPTER 2: LITERATURE REVIEW	10
2.1 INTRODUCTION	10
2.2 THE USAGE OF MICROSOFT EXCEL APPLICATION	10
2.2.1 THE USAGE AREA	11

2.3 A LOOK INTO EXCEL FILE FORMAT.....	12
2.4 JAVA TECHNOLOGY.....	12
2.4.1 THE JAVA PROGRAMMING LANGUAGE.....	13
2.4.2 THE JAVA PLATFORM.....	14
2.4.3 APPLYING JAVA TECHNOLOGY IN BUILDING OBJECT-ORIENTED DATA AUTOMATION SYSTEM.....	16
2.5 CONCEPT OF INTERFACING.....	17
2.5.1 JAVA NATIVE INTERFACE (JNI).....	17
2.5.2 CORBRA IDL.....	20
2.6 CALLING COM OBJECTS PROVIDED BY MICROSOFT USING JAVA TECHNOLOGY.....	23
2.6.1 OVERVIEW OF COMPONENT OBJECT MODEL (COM).....	23
2.6.2 PURPOSE OF CALLING COM OBJECTS.....	24
2.6.3 THE JACOB PROJECT – A JAVA-COM BRIDGE.....	25
2.6.3.1 USAGE AND DOCUMENTATION.....	28
2.7 SIMILAR PROJECT – MICROSOFT OFFICE AUTOMATION USING C++, J++ and Visual Basic.....	29
2.7.1 MICROSOFT WORD AUTOMATION USING C++.....	29
2.7.2 MICROSOFT EXCEL AUTOMATION USING VISUAL BASIC.....	30
2.7.3 MICROSOFT EXCEL AUTOMATION USING MICROSOFT J++.....	30
2.8 SUMMARY.....	31
CHAPTER 3: SYSTEM DEVELOPMENT METHODOLOGY.....	33
3.1 INTRODUCTION.....	33
3.2 OBJECT-ORIENTED METHODOLOGY.....	33
3.2.1 JUSTIFICATION OF CHOOSING OBJECT-ORIENTED TECHNIQUE.....	34
3.2.2 THE ADVANTAGES OF OBJECT-ORIENTED.....	35
3.3 OBJECT-ORIENTED TECHNIQUES.....	36
3.4 UNIFIED APPROACH (UA).....	37
3.4.1 PHASES IN UNIFIED APPROACH.....	38
3.4.2 OBJECT-ORIENTED ANALYSIS.....	39
3.4.3 OBJECT-ORIENTED DESIGN.....	40
3.4.4 INTERACTIVE DEVELOPMENT AND CONTINUOUS TESTING.....	40
3.4.5 UNIFIED APPROACH PROPOSED REPOSITORY.....	41
3.4.6 LAYERED APPROACH.....	42
3.5 UNIFIED MODELLING LANGUAGE.....	43
3.5.1 DEFINITION OF UML.....	44
3.5.2 JUSTIFICATION OF APPLYING UML.....	46
3.5.3 UML DIAGRAMS.....	47
3.6 SUMMARY.....	54
CHAPTER 4: OBJECT-ORIENTED ANALYSIS.....	55

4.1 INTRODUCTION	55
4.2 CURRENT PRACTICE	55
4.3 IDENTIFY ACTORS	56
4.4 BUSINESS PROCESS MODEL USING UML ACTIVITY DIAGRAM	57
4.4.1 CURRENT SYSTEM REPRESENTATION	58
4.4.2 PROPOSED SYSTEM REPRESENTATION	61
4.5 USE-CASE MODEL	64
4.6 INTERACTION BETWEEN SYSTEM AND USERS	65
4.7 CLASSES OF OBJECT IDENTIFICATION	69
4.7.1 IDENTIFICATION OF RELATIONSHIPS AMONG CLASSES	71
4.7.2 STATIC CLASS DIAGRAMS	73
4.8 SUMMARY	76
CHAPTER 5: OBJECT-ORIENTED DESIGN	77
5.1 INTRODUCTION	77
5.2 DESIGN AXIOMS AND COROLLARIES	77
5.3 DESIGNING CLASSES (ATTRIBUTES, METHODS, ASSOCIATIONS, STRUCTURES, PROTOCOLS)	80
5.3.1 REFINING ATTRIBUTES	81
5.3.1.1 REFINING ATTRIBUTES FOR THE EXCELSEPARATOR CLASS	82
5.3.1.2 REFINING ATTRIBUTES FOR THE EXCELCOMBINER CLASS	83
5.3.1.3 REFINING ATTRIBUTES FOR THE EXCELSEPARATORSERVICE CLASS	85
5.3.1.4 REFINING ATTRIBUTES FOR THE EXCELCOMBINERSERVICE CLASS	86
5.3.2 DESIGNING METHODS AND PROTOCOLS	87
5.3.2.1 DESIGNING METHODS FOR THE EXCELSEPARATOR CLASS	88
5.3.2.2 DESIGNING METHODS FOR THE EXCELCOMBINER CLASS	93
5.3.2.3 DESIGNING METHODS FOR THE EXCELSEPARATORSERVICE CLASS	97
5.3.2.4 DESIGNING METHODS FOR THE EXCELCOMBINERSERVICE CLASS	98
5.3.3 PACKAGES AND MANAGING CLASSES	99
5.4 DESIGNING ACCESS LAYER	101
5.5 DESIGNING USER INTERFACE	102
5.6 SUMMARY	105
CHAPTER 6: SYSTEM DEVELOPMENT	106

6.1 INTRODUCTION	106
6.2 SYSTEM DEVELOPMENT ENVIRONMENT	106
6.3 SOFTWARE DEVELOPMENT KIT	107
6.3.1 SYSTEM REQUIREMENTS	108
6.3.2 J2SDK INSTALLATION INSTRUCTIONS	109
6.4 SETUP PROCESS	110
6.4.1 TOMCAT SERVER SETUP PROCESS	111
6.5 DEVELOPMENT PROCESS	116
6.6 JAVA CODES TESTING PROCESS	120
6.7 CONCLUSION	121
CHAPTER 7: DEPLOYMENT AND TESTING	122
7.1 INTRODUCTION	122
7.2 DEPLOYMENT TO WEB SERVER	124
7.3 SYSTEM FUNCTIONALITY	126
7.3.1 SEGREGATING MICROSOFT EXCEL DOCUMENT	126
7.3.2 COMBINING MICROSOFT EXCEL DOCUMENTS	129
7.4 TESTING PROCEDURES	131
7.5 SUMMARY	141
CHAPTER 8: CONCLUSION	142
8.1 PROJECT STRENGTHS	142
8.2 PROJECT LIMITATION	143
8.3 SUGGESTION	144
8.4 CONCLUSION	145
REFERENCES	146
BIBLIOGRAPHY	148
APPENDIXES	149

LIST OF FIGURES

Figure No.	Title	Page
2.1	Excel File Format	12
2.2	Execution of a Java Program	14
2.3	NevNative.java	18
2.4	Compile NevNative.java	18
2.5	Create a C Header File	18
2.6	The NevNative.h File	19
2.7	The NevNativeImp.c File	19
2.8	Executing NevNative Class	20
2.9	Client-Server Communication Using CORBA	22
2.10	Using JACOB to Call COM Object	28
3.1	Phases in Unified Approach	39
3.2	Unified Modelling Language	44
3.3	Class Diagram	50
3.4	Use-Case Diagram	50
3.5	Sequence Diagram	51
3.6	Collaboration Diagram	51
3.7	Statechart Diagram	51
3.8	Activity Diagram	52
3.9	Component Diagram	52
3.10	Deployment Diagram	53
4.1	Activity diagram for Excel file segregation under current system practice	59
4.2	Activity diagram for Excel files combination under current system practice	60
4.3	Activity diagram for Excel file segregation – stage 1	61
4.4	Activity diagram for Excel file segregation – stage 2	62
4.5	Activity diagram for combining data	63
4.6	Use-Case Model	65
4.7	Sequence Diagram – Invalid Excel File Format	66
4.8	Sequence Diagram – Invalid Phrases	67
4.9	Sequence Diagram – Segregate Data in one Excel file	68
4.10	Sequence Diagram – Combine data in a few Excel files into one Excel File	69
4.11	The association relationship – between Interface objects and their processing objects	72
4.12	The association relationship – between Java Servlet objects and Java core objects	72
4.13	The association relationship – between Java core objects, it.bigatti.excel8 package and com.jacob.com package	72
4.14	The association relationship – between TempFileCreator object, ExcelSeparator object and FileDuplication object	72
4.15	The association relationship – between svtProcCombine	73

	Java Servlet object and svtAfterCombine Java Servlet object	
4.16	Static Class Diagram – ExcelCombiner class	73
4.17	Static Class Diagram – ExcelSeparator class	74
4.18	Static Class Diagram – ExcelCombinerService class	74
4.19	Static Class Diagram – ExcelSeparatorService class	74
4.20	Static Class Diagram – FileDuplication class	74
4.21	Static Class Diagram – TempFileCreator class	75
4.22	Static Class Diagram – svtSeparateExcel class	75
4.23	Static Class Diagram – svtCombineExcel class	75
4.24	Static Class Diagram – svtProcSeparate class	75
4.25	Static Class Diagram – svtProcCombine class	75
4.26	Static Class Diagram – svtAfterCombine class	75
5.1	The origins of the corollaries	80
5.2	Activity Diagram for modifyFile method	90
5.3	Activity Diagram for doSeparate method in ExcelSeparator class	92
5.4	Activity Diagram for run method in ExcelCombiner class	95
5.5	Activity Diagram for run2 method in ExcelCombiner class	96
5.6	Activity Diagram for doCombine method in ExcelCombiner class	97
5.7	The OODA System for Microsoft Excel Files Package	100
5.8	The Access Layer's Packages	102
5.9	User Interface for Calling ExcelSeparatorService Class	103
5.10	User Interface After Executing ExcelSeparatorService Class	104
5.11	User Interface for Calling ExcelCombinerService Class	104
5.12	The User Interface after Executing ExcelCombinerService Class	105
6.1	J2SDK File Structure	109
6.2	Tomcat startup Command	112
6.3	Tomcat startup Successfully	113
6.4	Tomcat shutdown Command	114
6.5	Testing Tomcat Server	115
6.6	Process Involved in Creating Java file and Class file	117
6.7	Creating Directory in Tomcat Servlet Application	119
7.1	File Structure of the System Deployment	123
7.2	Process Flow for System Deployment Process	125
7.3	Screen for Segregating an Excel File Document	127
7.4	Screen for Combining Microsoft Excel Documents	130
7.5	The Form for Recording the Testing Process of Segregating Microsoft Excel Files	133
7.6	The Form for Recording the Testing Process of Combining Microsoft Excel Files	134
7.7	The Form for Recording the Comments and Suggestions from System Tester on the OODA System for Microsoft	135

	Excel Files	
7.8	Testing Procedures for the OODA System for Microsoft Excel Files	140
E1	Microsoft Excel 2000 Object Model	174
F1	Startup menu screen	178
F2	Segregating Excel document screen	178
F3	Browse folder screen	179
F4	Segregating Excel result screen	179
F5	Combining Excel documents screen	180
F6	Combining Excel result screen	180
G1	Example Excel file for segregating purpose	181
G2	Example of segregated result : Part 1	182
G3	Example of segregated result : Part 2	183
G4	Example of segregated result : Part 3	184
H1	Example Excel file1 for combining purpose	185
H2	Example Excel file2 for combining purpose	185
H3	Example Excel file3 for combining purpose	185
H4	Result of vertically combine	186
H5	Result of horizontally combine	186

LIST OF TABLES

Table No.	Title	Page
2.1	The Java-COM Bridge Providers	25
3.1	Dynamic Diagrams	49
4.1	Two Main Roles of the Actor – Excel File Administrator	57
4.2	Use-Case Names and their Descriptions	64
4.3	Executable Use-cases	70
4.4	Types of Object and their specific classes/packages of object	71
5.1	O-O Design Axioms	78
5.2	O-O Design Corollaries	79
5.3	Attributes for the ExcelSeparator Class Identified during O-O Analysis Stage	82
5.4	Redefined Attributes for the ExcelSeparator Class	83
5.5	Attributes for the ExcelCombiner Class Identified during O-O Analysis Stage	84
5.6	Redefined Attributes for the ExcelCombiner Class	85
5.7	Redefined Attributes for the ExcelSeparatorService Class	86
5.8	Redefined Attributes for the ExcelCombinerService Class	86
5.9	Types of methods of a class	87
6.1	Operating Systems Accepted by J2SDK	108
6.2	Tomcat Server Installation Steps	111
6.3	Testing Java Servlet Pages	114
6.4	Differences between a Java Servlet page and a normal Java program	118
6.5	Tomcat Configuration Steps	120
7.1	Validation Involved in System Function – Segregating Microsoft Excel Document	129
7.2	Validation Involved in System Function – Combining Microsoft Excel Documents	131
7.3	Demography Analysis of System Tester	136
7.4	Testing Results – Segregating Excel Document	137
7.5	Testing Results – Combining Excel Documents	138
7.6	Overall Testing Results	139
D1	Test System Functionality: Segregating Excel document	168
D2	Comments Gathered for System Functionality – Segregating Excel Document	169
D3	Test System Functionality: Combining Excel documents	170
D4	Comments Gathered for System Functionality – Combining Excel Documents	171
D5	Overall Comments and Suggestions	172
E1	Representation of Object and Collection for Microsoft Excel 2000 Object Model	173

LIST OF ABBREVIATION

ICT	Information Communication Technology
OODA	Object-Oriented Data Automation
COM	Component Object Model
JVM	Java Virtual Machine
GUI	Graphical User Interface
API	Application Programming Interface
TCP	Transmission Control Protocol
IP	Internet Protocol
UDP	User Datagram Protocol
JDBC	Java Database Connectivity
JNI	Java Native Interface
CORBA	Common Object Request Broker Architecture
ORB	Object Request Broker
IDL	Interface Definition Language
OLE	Object Linking Embedding
GUID	Globally Unique Identifier
DLL	Dynamic Link Library
JACOB	Java-COM Bridge
UA	Unified Approach
UML	Unified Modelling Language
OMT	Object Modelling Technique
OOBE	Object-Oriented Business Engineering
OOSE	Object-Oriented Software Engineering
OOA	Object-Oriented Analysis
RAM	Random Access Memory
DBMS	Database Management System
SQL	Structured Query Language
RDBMS	Relational Database Management System
J2SDK	Java Software Development Kit Version 2

LIST OF APPENDIXES

Appendix	Title	Page
A	A usage scenario for Microsoft Word automation program by using the CautoWord class	145
B	Microsoft Excel Automation using Visual Basic	157
C	Microsoft Excel Automation using Microsoft Visual J++	165
D	System testing result	168
E	Microsoft Excel Objects	173
F	Screen shots	178
G	Example of Excel files – Segregating function	181
H	Example of Excel files – Combining function	185
I	Source code – ExcelCombiner.java	187